

**BRAGBOARD PROJECT WEEK 3 REPORT:-**

**NAME : CHITIRALA GIRIDHAR**

**GMAIL: GIRIDHARCHITIRALA@GMAIL.COM**

**GITHUB LINK:- <https://github.com/springboardmentorx5768/Project-2.git>**

### **Week 3 Report: Shout-Out Module with Recipient Tagging**

#### **Objectives:-**

- Design and implement a **shout-out form** for users to appreciate peers.
- Enable **tagging recipients** using dropdown or search.
- Store shout-out messages and tagged users in a **database**.
- Handle form validation, backend integration, and error handling.

#### **Key Components & Terminology**

<b>Term</b>	<b>Explanation</b>
<b>Shout-out Form</b>	A frontend UI component where users can write a message and tag recipients.
<b>Recipient Tagging</b>	Feature allowing users to select one or more teammates (e.g., via dropdown or autocomplete) to receive the shout-out.
<b>Database (DB)</b>	A backend system (e.g., PostgreSQL, MongoDB) used to store shout-out messages and recipient data.
<b>API Endpoint</b>	A backend route (e.g., POST /api/shoutout) that receives form data and stores it in the DB.
<b>Form Validation</b>	Ensuring required fields (message, recipients) are filled before submission.
<b>Schema Design</b>	Structure of the database tables/collections to store shout-outs and user references.
<b>Foreign Key</b>	A field in one table that links to the primary key of another (e.g., shout-out → user).

## **Implementation Overview :-**

### **1. Frontend (React + Formik + Material UI)**

- **Fields:**
  - message: TextArea for the shout-out.
  - recipients: Multi-select dropdown with user names (fetched from backend).
- **Validation:**
  - Required: message and at least one recipient.
- **Submission:**
  - Sends POST request to backend with { message, recipientIds }.

### **2. Backend (FastAPI + SQLAlchemy)**

- **Models:**
  - User: Stores user info.
  - ShoutOut: Stores message content and timestamp.
  - ShoutOutRecipient: Many-to-many table linking shout-outs to recipients.
- **API Endpoint:**
  - POST /shoutout/: Accepts JSON payload, validates user IDs, stores data.

### **3. Database (PostgreSQL)**

- **Tables:**
  - users(id, name, email)
  - shoutouts(id, message, created\_at)
  - shoutout\_recipients(id, shoutout\_id, user\_id)

## Common Errors & Fixes

Error	Cause	Fix
<b>400 Bad Request</b>	Missing required fields in form submission	Added Formik validation schema using Yup
<b>500 Internal Server Error</b>	Incorrect DB schema or missing foreign key	Updated SQLAlchemy models and added proper relationships
<b>Recipient not found</b>	Invalid user ID in request	Added backend check to validate user existence before insertion
<b>CORS Error</b>	Frontend and backend on different ports	Enabled CORS middleware in FastAPI
<b>Empty recipient list</b>	User submits form without selecting recipients	Added frontend validation to enforce at least one recipient

## Learnings & Improvements

- Understood how to implement **many-to-many relationships** in SQLAlchemy.
- Practiced **form validation** using Formik + Yup.
- Improved debugging skills by reading **FastAPI error logs**.
- Learned to use **Postman** for API testing before frontend integration.

PICTURES OF BRAG BOARD WEEK 3:

The screenshot shows the BragBoard application interface. On the left, there is a sidebar with a user profile picture and the name "shannuk Employee". Below the profile are navigation links: Dashboard, Shout-outs (which is the active tab), Achievements, Goals, and Settings. The main content area has a title "Shout-outs". It contains a "Share a Shout-out" form with a text input field ("Give recognition to someone who made a difference..."), a "Tag People" button, an "Upload Image" button, and a large blue "Post Shout-out" button. Below this is a "Filters" section with dropdowns for "Search by sender...", "All Departments", and "All Time", and a "Reset Filters" button. A message "Feed (0 shout-outs)" is displayed, followed by "Loading shout-outs...". At the bottom of the screen, there is a taskbar with various icons and system status information.

### Shoutout layout :-

This screenshot shows the same BragBoard interface as above, but with a department filter applied. In the "Filters" section, the "All Departments" dropdown is open, showing a list of departments: All Departments (selected), Engineering, Human Resources, Marketing, Sales, Operations, and General. The "Feed (19 shout-outs)" section shows two posts: one from "shannuk" and another from "hlo", both tagged to "Giridhar Chitrala". The taskbar at the bottom remains the same.

### Department wise scopeing:-

This screenshot shows a detailed view of a shout-out post from "shannuk". The post content is "hlo". Below the post, it says "Tagged: Giridhar Chitrala • employee". At the bottom of the post view, there are three small icons: a heart, a star, and a star with a plus sign. The taskbar at the bottom is visible.

Feed of shout-out:-

### Explanation of the Uploaded Shout-Out Feed

The image shows a **live feed** of shout-outs posted by users on your platform. Here's a breakdown of the visible elements:

#### UI Elements

##### Component    Description

<b>Feed Header</b>	Shows "Feed (19 shout-outs)" indicating total shout-outs posted.
<b>Shout-Out Card</b>	Each post includes sender name, role (e.g., "employee"), message, timestamp, and tagged recipients.
<b>Message</b>	Example: "hlo" posted by <b>shanmuk</b> 3 days ago.
<b>Tagged Section</b>	Shows <b>Girdhar Chintala</b> as the tagged recipient.
<b>Icons</b>	Like  , Comment  , and Share  buttons for interaction.

#### Backend Mapping

This UI reflects successful integration of your backend shout-out module:

- **Sender Info:** Retrieved from users table.
- **Message:** Stored in shoutouts table.
- **Tagged Users:** Retrieved via shoutout\_recipients join table.
- **Timestamp:** Auto-generated on shout-out creation.
- **Interactions:** Can be linked to separate tables for likes/comments if extended.

#### Validation Success

This feed confirms:

- Form submission is working.
- Tagged users are stored and retrieved correctly.
- UI is rendering dynamic data from the database.

### **Suggested Enhancements**

- **Message Length Check:** "hlo" is very short—consider enforcing a minimum character count.
- **User Avatars:** Add profile pictures for better visual identity.
- **Tagging UI:** Show multiple tagged users with chips or badges.
- **Interaction Counts:** Display number of likes/comments.

This all created and updated in shout-out in week 3  
this is the report of week 3

Thank you for guide me mam,...